**Project Abstract:**

This project applies **K-Means clustering** to segment customers of a mall based on their **age**, **annual income**, and **spending score**. Using unsupervised learning, the model groups customers into distinct segments, allowing for targeted marketing strategies and personalized services. The project explores multiple cluster numbers, evaluates model performance using silhouette scores, and identifies six optimal customer groups. This segmentation provides insights into consumer behavior, helping businesses to enhance customer engagement and optimize marketing efforts.

**Why Use Customer Segmentation in "MallCustomerSeg":**

Customer segmentation in the **MallCustomerSeg** project helps to group customers based on their spending behavior, age, and income. It allows businesses to understand different customer groups and tailor marketing strategies accordingly. Segmentation can identify high-spending customers, frequent visitors, or those with potential for upselling. This approach enhances personalized marketing, improves customer satisfaction, and drives business growth.

**Algorithm:**

* **K-Means Clustering** is used, a popular unsupervised machine learning algorithm for partitioning data into distinct groups or clusters based on similarity.

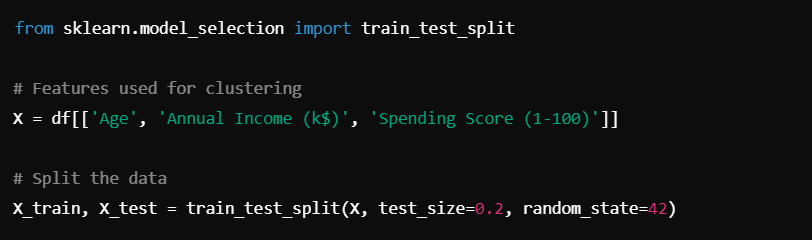
**Steps in the Notebook:**

1. **Feature Selection**: The relevant features for clustering are:
   * **Age**
   * **Annual Income (k$)**
   * **Spending Score (1-100)**
2. **Scaling the Features**: The data is likely scaled using StandardScaler to ensure all features contribute equally to the distance calculations in the clustering process.
3. **Choosing the Number of Clusters**:
   * The notebook explores different values for **k** (the number of clusters) using a range (2 to 12) and calculates the **inertia** (sum of squared distances between points and their assigned cluster's centroid) for each. Lower inertia generally indicates better clustering.
   * It also tests for the best **silhouette score**, which measures how similar an object is to its own cluster compared to other clusters. A higher score indicates better-defined clusters.
4. **Model Training**:
   * The final model is trained with **6 clusters** (KMeans(n\_clusters=6)), based on the exploration of cluster numbers.
   * Each customer is assigned a cluster label, indicating their segment.
5. **Saving the Model**: The trained KMeans model is saved using joblib for later use or deployment.

**To simulate a train-test split:**

If you still wish to split the data for testing, you can do the following:

1. **Split the Data**: Use train\_test\_split from sklearn to divide the data, even though clustering typically doesn’t require this.



**2.Fit the Model on Training Data**:

**kmeans = KMeans(n\_clusters=6, random\_state=42)**

**kmeans.fit(X\_train)**

**3. Evaluate on Test Data**:

You can then test how well the clustering generalizes by assigning clusters to the test data:

**predictions = kmeans.predict(X\_test)**